Project of Research Assignment for Exchange Student

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona (ETSEIB)
Research Centre for Biomedical Engineering(CREB)

System Identification from Physiological Signals Using Black-box Modelling Concepts

01.09.2019 - 31.01.2020

Author: Genki Imamura and Natsuki Igarashi

Director: Montserrat Vallverdu

ETSEIB

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

UPC

Contents

1 Introduction

System identification is a methodology for building mathematical models of dynamic systems using measurements of the system's input and output signals.

The process of system identification requires work described below.

First, measure the input and output signals from your system in time or frequency domain. Next, select a model structure. Apply an estimation method to estimate value for the adjustable parameters in the candidate model structure. And then, evaluate the estimated model to see if the model is adequate for your application needs.In particular, this process is called validation process.

Black-box models are the functional relationships between system inputs and system outputs. In a dynamic system, the values of the output signals depend on both the instantaneous values of its input signals and also on the past behavior of the output system.

In this project, we identify the system from the HRV (Heart Rate Variability) obtained of several ECG (Electrocardiogram) . We used MATLAB for analyzing the data. ECG is the physiological signal and it is the recording of electrical activity of the myocardium of the heart during each cardiac cycle. ECG wave forms consist of each waves of P , QRS , T , U.

We use the ECG data from PhysioNet. This web site provides a lot of physiological signal data as open source. We attempt to obtain RR intervals (RRI) from these data.

We have written MATLAB scripts in order to codify the models and also we have used the identification MATLAB app.

Setting the input and output in matlab app and detrend, select polynomial model and try several models.

We set as input random noise and set as output the RRI.

After we obtained the models, we can compare its errors and how the model fits the signal.

ETSEIB

2 Physiological terms used in this project

To start our project, we learned about physiological terms. These terms are described below.

ECG①

ECG stands for electrocardiogram. This is kind of an electrical wave or signal of medication. As observating this wave, someone can find whether there is an arrhythmia, an angina, a myocardial infarction.

Arrhythmia②

Arrhythmia is also called irregular heartbeat. That is a problem with someone's heart, including faster or slower than normal. And more symptoms of arrhythmia have skipping beats, lightheadedness or dizziness, chest pain, shortness of breath and sweating. Arrhythmia includes **Supraventricular**, **Premature**, **Ventricular** and **Conduction Defect** as irregular symptoms.

Pacemaker③

A pacemaker is a small device which is used to treat arrhythmia. The hearts has own internal electrical systems that control the rhythm of heartbeat. So the pacemaker help the heart to move normally by electricity.

Heart rate variability parameters④⑤⑥

In this project, the following terms are commonly used as heart rate variability(**HRV**) parameters and should be mastered. **RR** stands for respiratory rate which is possible to confirm in ECG signals as between peak to peak. **HR** is a mean value of total heart rate for 5 minutes. **SDNN** is standard deviation of RR interval for 5 minutes. **ST** is the part of the ECG from the end of the S-wave to the beginning of the T-wave. It is used to diagnose angina, myocardial infarction, pericarditis, or ventricular hypertrophy.

References

① Japan Society of Ningen Dock ,

https://www.ningen-dock.jp/public/inspection/electrocardiogram

② MedinePlus , https://medlineplus.gov/arrhythmia.html

③ NIH,What is a pacemaker , https://www.nhlbi.nih.gov/health-topics/pacemakers

④ Japan Heart Foundation, About ST, https://www.jhf.or.jp/check/term/word_a/st/

⑤ European Heart Journal (1996) 17, 354–381, Heart rate variability, Introduction y Table 1

⑥ The ST Segment, LIFE IN THE FASTLINE ehttps://litfl.com/st-segment-ecg-library/

ETSEIB

3 Analyzed databases

3.1 Physionet

We have downloaded ECG data from Physionet [1] [4] in order to obtain RR signals using MATLAB algorithm. Once, the RR signals will be obtained they are going to be preprocessed.

As following, we installed data and prepared to use these data in MATLAB.

1. Install WFDB

   Launching MATLAB and installing The native Python waveform-database (WFDB)[2] which is reading, writing, and processing WFDB signals and annotations. As bellow, showing the code to install WFDB:

```
[old_path]=which('rdsamp');

if(~isempty(old_path))
rmpath(old_path(1:end-8));

end wfdb_url='https://physionet.org/
physiotools/matlab/wfdb-app-matlab/wfdb-app-toolbox-0-10-0.zip';
[filestr,status] =
urlwrite(wfdb_url,'wfdb-app-toolbox-0-10-0.zip');
unzip('wfdb-app-toolbox-0-10-0.zip');

cd mcode
addpath(pwd)
savepath
```

2. Obtaing the ECG Data from Physionet[1]

   We obtained HRV data from three patients from Physionet[1] which includes three types of file (.atr, .dat and .hea). In this project, we used patients numbered with 100, 101 and 103.

3. Saving M data

   We saved M data that is .m to MATLAB/mcode. And then we could write the M file's name (ECGplot) into the command window or Explorer.

4. Reading data

   After preparing the data that is someone's ECG data, we read data by using rdsamp as showing:

```
[ecg,Fs,tm]=rdsamp('mitdb/100',1);
```

   ecg means amplitude data of ECG. That is N*1 of double that N is the number of the samples. **Fs** is sampling frequency and **tm** is like ecg vector but this is representing the sampling time.

ETSEIB

Next, we read ECG data by rdann as showing:

```
[ann,type,subtype,chan,num]=rdann('mitdb/100','atr',1);
```

ann means integer vector of annotation locations of R wave determined by cardiologists. This vector is also N*1.

5. Wavelet transform

   We used wavelet transform down to level 5 as **wt** for decomposing ECG waveform. Simultaneously we used only the wavelet coefficients at scales 4 and 5 with 11.25Hz to 22.50Hz and 5.625Hz to 11.25Hz for localizing waveform. And we reconstructed a frequency-localized version of the ECG waveform using the default **sym4** wavelet:

```
wt =modwt(ecg,5);
wtrec = zeros(size(wt));
wtrec(4:5,:) = wt(4:5,:);
y = imodwt(wtrec,'sym4');
```

6. Derivation of R wave

   We used the squared absolute values of the signal approximation built from the wavelet coefficients and peak finding algorithm to identify the R peaks as showing:

```
y = abs(y).^2;
[qrspeaks,locs] = findpeaks(y,tm,'MinPeakHeight',0.22,
                   'MinPeakDistance',0.150);
```

   qrspeaks is local maxima and **locs** is Peak locations. **MinPeakHeight** is Minimum Peak Height and **MinPeakDistance** is Minimum Peak Distance.

7. ECG plot

   We are plotting the data for showing the graphs in this part.

   It is plotting row data. **xlim** and it is setting the x-axis limits for the chart:

```
figure
plot(tm,ecg) %ecg
xlim([0 15])
```

   These are plotting wavelet reconstruction data and ann. And **hold on** and **hold off** are the commands that are able to overlap something to executable statement:

```
hold on
plot(tm,y,'r') %Wavelet Reconstruction
plot(tm(ann),ecg(ann),'ro') %ann
hold off
```

   This is labeling the chart elements for some graphs:

```
title('ECG')
xlabel('Seconds')
ylabel('Amplitude')
legend('Raw Data','Wavelet Reconstruction','Ann
Data','Location','SouthEast');
```

8. Derivation of R-R interval (RRI)

   To compare between wavelet and ann, we obtained time series of annotation data from annotation locations. And then, we introduce vector of RRI (wavelet) to variable rrint. Likewise, we introduce vector of RRI (annotation) to variable annint:

```
anntm = tm(ann);
for k=1:length(locs)-1
    rrint(k) = locs(k+1) - locs(k);
    annint(k) = anntm(k+2) - anntm(k+1);
end
```

9. RRI plot (Comparison of wavelet and ann)

   Plotting RRI obtained from wavelet Reconstruction data. **bar** is plotting bar graph:

```
figure
subplot(1,2,1)
bar(rrint,0.4,'b')
```

   Labeling for wavelet filter:

```
title('RR interval(wavelet filter)')
xlabel('Number of intervals')
ylabel('RRI')
```

   Plotting RRI obtained from Ann data:

```
hold on
subplot(1,2,2)
bar(annint,0.4,'r')
```

   Labeling for annotation:

```
title('RR interval(ann)')
xlabel('Number of intervals')
ylabel('RRI')
hold off
```

That is obtaining the RR's length of time:

```
locs2 = locs(1:length(locs)-1);
```

locs2 means RR's intervals length.

Plotting and labeling RRI for wavelet:

```
Figure
plot(locs2,rrint)
title('RR interval(wavelet filter)')
xlabel('time')
ylabel('RRI')
```

And now it is plotting difference between wavelet and ann:

```
Figure
difference = abs(rrint - annint);
bar(difference,0.4)
title('Difference between wavelet filter and ann')
xlabel('Number of intervals')
ylabel('Diffrence')
```

3.2 FLUKE ProSim2

ProSim is the equipment that simulates biological information of patients. As we analysing some patients ECG waves, we are able to obtain by using it.

Actually, we could not analyze the data from ProSim because of necessity of cable to monitor and obtain these signals. Now on we obtain the data from physionet[1]. This will be described other section. In this section, we describe about FLUKE ProSim2.

First of all, we need to set and check the parameters about conditions of patients, environment and so on. And we can see the screen for setting. In the home-screen, it is able to set the category as below. In addition, the name in parentheses "( )" indicates the titles on the screen.

- Cardio Output (CO)
- ECG
- Respiration (RESP)
- Blood Pressure (BP)
- Arrhythmias (ARRY)
- PerformanceWave (PERF)
- PacemakerWave (PACE)
- Temperature (TEMP)
- FetalMaterial (FE/MA)

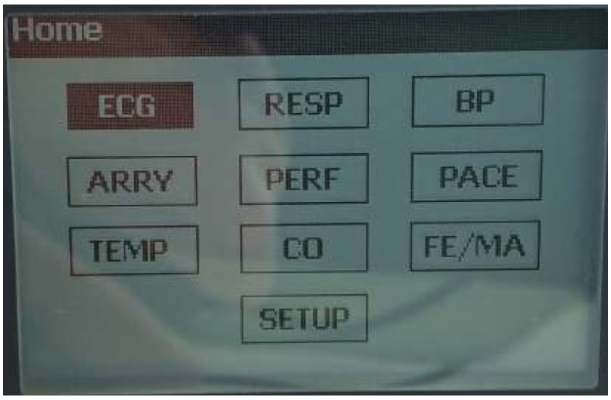

Figure 1: FLUKE Home Screen

Arrhythmia In this option, we can set the type of the arrhythmia. That is including Supraventricular, Premature, Ventricular and Conduction Defect.

ECG In this ECG option, we can set as bellow table1.

Table 1: Option for ECG

Rate	Min. 30[bpm] Max. 300[bpm] steps by 20[bpm]
Ampl	Min. 0.05[mV] Max. 5.50[mV] steps by 0.5[mV]
PT TYPE	ADULT or PETS
ST	Min. -0.80[mV] Max. 0.8[mV] steps by 0.1[mV]
Aref	off, Resp, Wandr, Muscle, 60Hz, 50Hz

Rate is heart rate of patients that is number of beats per minute and **Ampl** is amplitude of that. **Aref** is "Artifact". **Resp** is "Respiration", **Wandr** is wandering baseline.

4 Methodology

4.1 Principle of black-box

We refer to the document that is Dynamics Linear Systems Identification MATLAB Practices[3]

The general structure of a linear system

$$y(t) = G(q)\,u(t) + H(q)\,e(t) \tag{1}$$

$u(t) \cdots$ input, $y(t) \cdots$ output

$e(t) \cdots$ white noise, $t \cdots$ instants of sampling

Parametric estimation methods

It is possible to parametrized $G(q)$ and $H(q)$, representing them as rational functions, being its parameters θ the numerator and denominator coefficients.

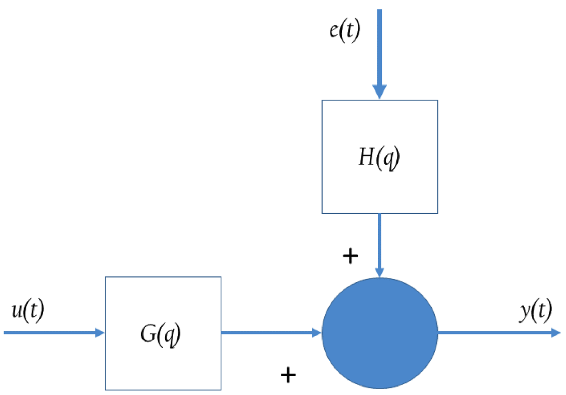

Figure 2: Block diagram of the general structure of a linear system

The functions $G(q)$ and $H(q)$ give place to different classes of models according to the type of terms that might be presented. As a result, this gives place to a models family, for example: FIR (Finite Impulse Response), ARX (Auto Regressive with eXogenous inputs), ARMAX (Auto Regressive Moving Average with eXogenous inputs), etc. The whole model family can be analytically represented by the expression:

$$A(q)\,y(t) = \frac{B(q)}{F(q)}\,u(t) + \frac{C(q)}{D(q)}\,e(t) \tag{2}$$

And the block diagram representation is shown in the Figure3.

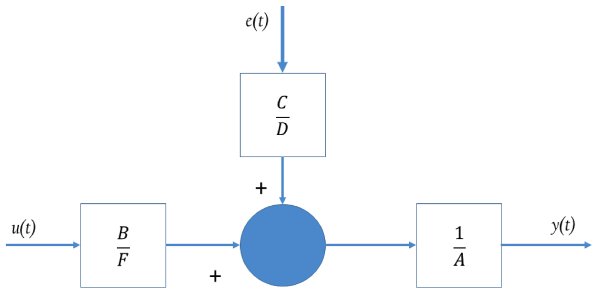

Figure 3: Block diagram of the whole model family expression

$$
\begin{aligned}
A &= A(q) = 1 + a_1 q^{-1} + ... + a_{na} q^{-na} \\
B &= B(q) = q^{-nk} B(q) \\
B(q) &= b_0 + b_1 q^{-1} + ... + b_{nb} q^{-nb+1} \\
C &= C(q) = 1 + c_1 q^{-1} + ... c_{nc} q^{-nc} \\
D &= D(q) = 1 + d_1 q^{-1} + ... + d_{nd} q^{-nd} \\
F &= F(q) = 1 + f_1 q^{-1} + ... + f_{nf} q^{-nf}
\end{aligned}
\tag{3}
$$

being

$$q^{-1}$$

the delay operator:

$$q^{-1} u(t) = u(t-1)$$

One characteristic that differences the structures that are part of the general equation (1), is the modeling form of the stochastic or noise part. For this reason, the models are grouped in two blocks by the polynomials that characterizing $H(q)$, different models are shown below.

	Polynomials Utilized	Model Name
$H(q) = 1$	BF	OE
$H(q) \neq 1$	AB	ARX
	ABC	ARMAX
	BFCD	BJ

4.1.1 ARX Model

This model can be described with a block diagram, as showing in the Figure4.

An ARX (Auto Regressive with eXogenous inputs) model has the next temporary relation between the input and the output:

$$
\begin{aligned}
y(t) + a_1 y(t-1) + a_2 y(t-2) + ... + a_{nb} y(t - n_{nb}) = b_1 u(t-1) \\
+ b_2 u(t-2) + ... + b_{nb} u(t - n_{nb}) + e(t)
\end{aligned}
\tag{4}
$$

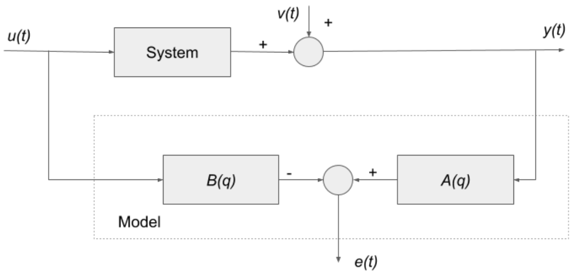

Figure 4: Block diagram of a ARX model

This differences equation can be written trough the following transfer function as showing Figure3:

$$G(q) = \frac{B(q)}{A(q)} \tag{5}$$

$$H(q) = \frac{1}{A(q)} \tag{6}$$

And then the vector with the perturbation values of the system v(t) is correlated with the prediction error e(t) as showing Figure4:

$$e(t) = A(q)v(t) \tag{7}$$

The $C(q)$ function is called MA (moving average) and it is the part which model the prediction residue, as the result of a linear filter applied to a white noise sequence.MA (moving average) model, so that $A(q) = 0$, $B(q) = 0$ and $C(q) \neq 0$.

4.1.2 ARMAX Model

This model can be described with a block diagram, as showing in the Figure 5. An ARMAX

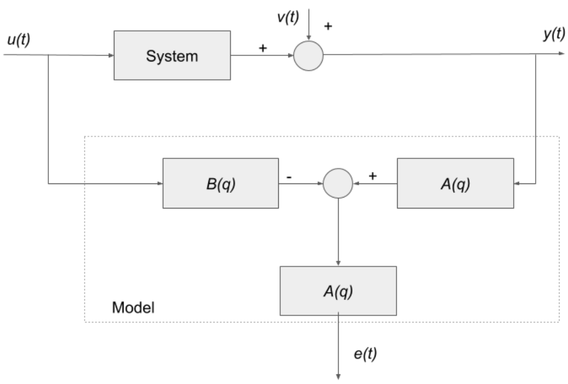

Figure 5: Block diagram of a ARMAX model

structure responds to the next expression:

$$\begin{aligned} y(t) + a_1 y(t-1) + a_2 y(t-2) + \ldots + a_{nb} y(t-n_{nb}) = b_1 u(t-1) + b_2 u(t-2) \\ + \ldots + e(t) + c_1 e(t-1) + \ldots + c_{nc}(t-n_{nc}) \end{aligned} \tag{8}$$

The transfer functions resulted are:

$$G(q) = \frac{B(q)}{A(q)} \tag{9}$$

$$H(q) = \frac{C(q)}{A(q)} \tag{10}$$

as showing figure2

4.2 Parametric models using MATLAB script

This is the example of model identification used by MATLAB script below[5] :

```
>> present(ARXsim101)

ARXsim101=Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)
A(z) = 1 - 0.8376 (+/- 0.023) z^-1 + 0.1362 (+/- 0.03) z^-2
-0.0372 (+/- 0.02998) z^-3 - 0.1456 (+/- 0.02425) z^-4

B(z) = 0.139 (+/- 0.02268) z^-1 - 0.09042 (+/- 0.02743) z^-2
-0.148 (+/- 0.02733) z^-3 + 0.1488 (+/- 0.02169) z^-4

Name: ARXsim101
Sample time: 1 seconds

Parameterization:
    Polynomial orders:    na=4    nb=4    nk=1
    Number of free coefficients: 8
    Use "polydata", "getpvec", "getcov" for parameters
    and their uncertainties.

Status:
Estimated using ARX on time domain data "mydatad".
Fit to estimation data: 13.85%  (simulation focus)
FPE: 0.003523, MSE: 0.003558
More information in model's "Report" property.
```

Syntax **present()** displays the linear or nonlinear identified model and the following information:

- Estimated one standard deviation of the parameters, which gives 68.27 % confidence region
- Termination conditions for iterative estimation algorithms
- Status of the model — whether the model was constructed or estimated
- Fit to estimation data
- Akaike's Final Prediction Error (FPE) criterion
- Mean-square error (MSE)

FPE (Akaike's Final Prediction Error) criterion provides a measure of model quality by simulating the situation where the model is tested on a different data set. And it can compare them using this criterion. The most accurate model has the smallest FPE.

MSE (mean squared error) is an effective method to quantify the performance of a linear regression model, and it is the average of the error between the actual value and the predicted value by the model.

4.3 Parametric models using MATLAB APP

To open the System Identification app, type the following command in the MATLAB script [6] :

```
>>ident
```

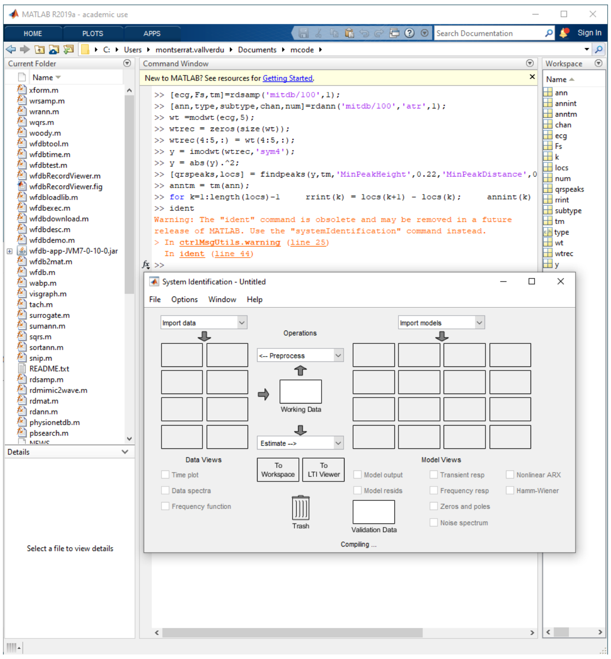

Figure 6: Open the System Identification app on MATLAB

And then, we import the input and output signals into the app from the MATLAB workspace.

To import data arrays into the System Identification app. Open the **Import data** dialog box.

Select **Import data > Time domain data.**

The **Import data** dialog box now resembles the following figure.

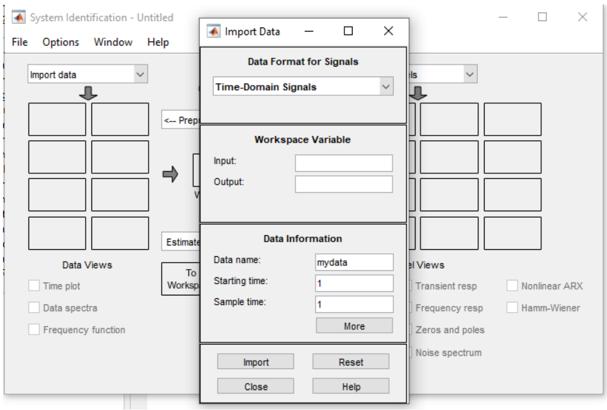

Figure 7: Import data dialog box

In the **Import data** dialog box, specify the following options:

- Input
- Output
- Data name — Change the default name to data. This name labels the data in the System Identification app after the import operation is completed.
- Starting time — This value designates the starting value of the time axis on time plots.
- Sample Time

In the **Data Information** area, click **More** to expand the dialog box and it can specify the following options:

In this project,we use random input generated by the script below.

```
rand=1.13-0.6.*rand(1,2272);
```

(1,2272) is the number of matrix of RRI,we have to change it if the another signals set.

Input Properties

- InterSample — Accept the default zoh (zero-order hold) to indicate that the input signal was piecewise-constant between samples during data acquisition. This setting specifies the behavior of the input signals between samples when you transform the resulting models between discrete-time and continuous-time representations.

- Period — Accept the default inf to specify a nonperiodic input.

Click **Import** to add the data to the System Identification app.

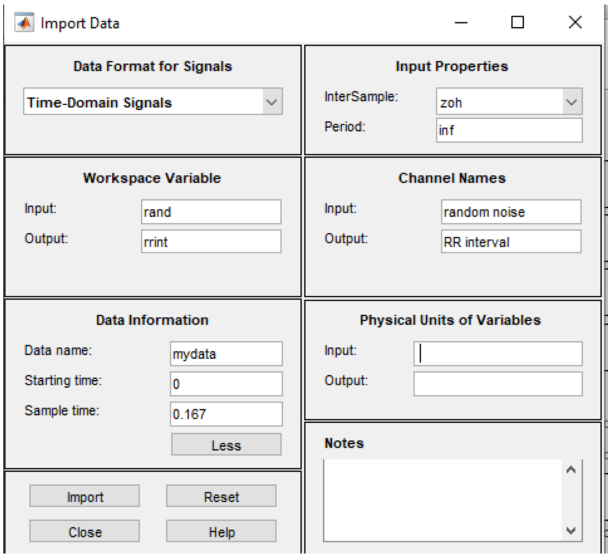

Figure 8: Import the time domain data

Click **Close** to close the Import Data dialog box.

Remove the mean input value from the input data and the mean output value from the output data.

In the System Identification app, select **<–Preprocess> Remove means**.

This action adds a new data set to the System Identification app with the default name **datad** (the suffix d means detrend), and updates the **Time Plot** window to display both the original and the detrended data.

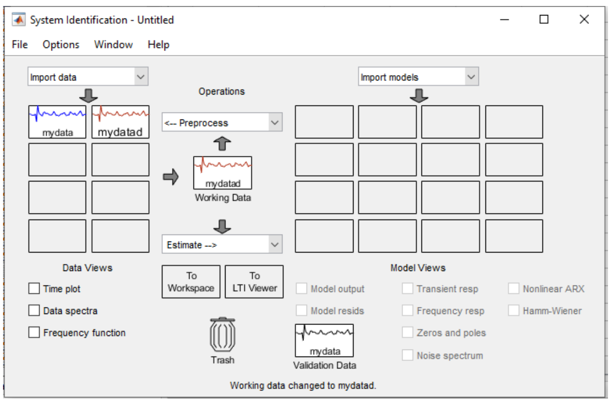

Figure 9: Remove the mean value

The detrended data has a zero mean value.

Drag the data set **datad** to the **Working Data** rectangle to specify the detrended data to be used for estimating models.

In addition, to click the **Time plot** check box, we can see the time plot data.

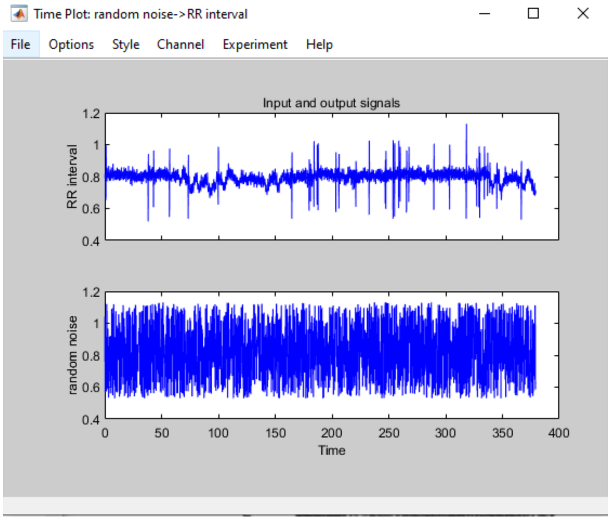

Figure 10: Time plot data

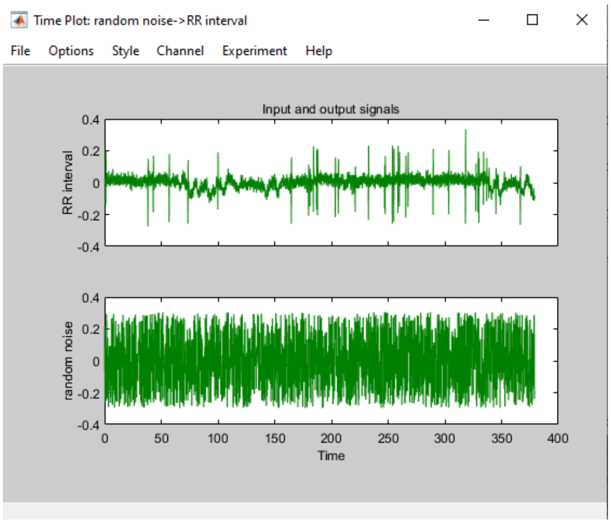

Figure 11: Detrended data

To get information about a data set, right-click on it icon to open the **Data/model** Info dialog box.

Explanation of the following values in the **Data/model** Info dialog box.

- Changing the name of the data set in the **Data name** field.
- Changing the color of the data icon in the **Color** field. You specify colors using RGB values (relative amounts of red, green, and blue). Each value is between 0 and 1. For example, $[1, 0, 0]$ indicates that only red is present, and no green and blue are mixed into the overall color.

- Viewing or editing the commands executed on this data set in the **Diary and Notes** area. This area contains the command-line equivalent to the processing you performed using the System Identification app.

Open the **Estimate –>** dialog box and choose the **Polynomial Models**, **Polynomial Models** window appeared.

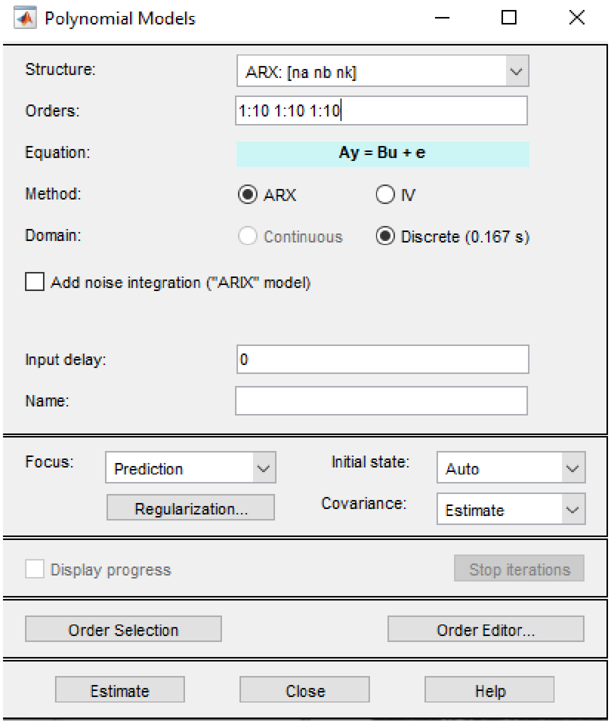

Figure 12: Select the polynomial models

The model-output plot shows the model response to the input in the validation data.

The fit values for each model are summarized in the **Best Fits** area of the Model Output window.

The models in the **Best Fits** list are ordered from best at the top to worst at the bottom. The fit between the two curves is computed such that 100 means a perfect fit, and 0 indicates a poor fit (that is, the model output has the same fit to the measured output as the mean of the measured output).

From the Structure list, select ARX: [na nb nk].

Edit the Orders field to try all combinations of poles, zeros, and delays, where each value is from 1 to 10:[1:10 1:10 1:10]

Click **Estimate** to open the **ARX Model Structure Selection** window, which displays the model performance for each combination of model parameters.

We use this plot to select the best-fit model.

- The horizontal axis is the total number of parameters — na + nb.

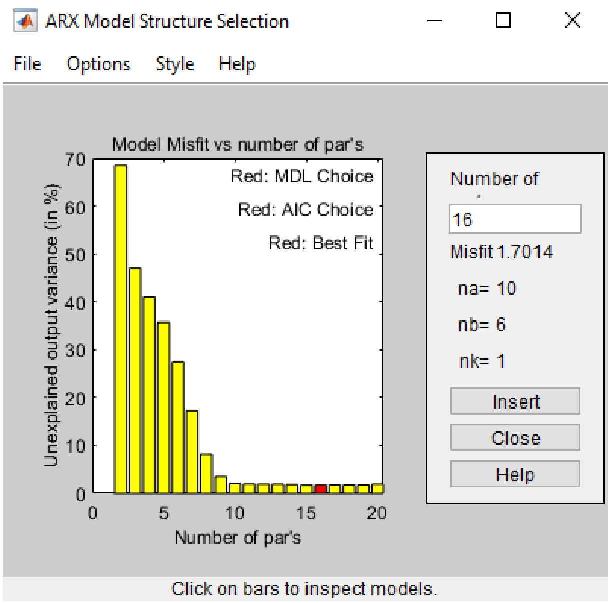

Figure 13: Model structure selection

- The vertical axis, called Unexplained output variance (in %), is the portion of the output not explained by the model—the ARX model prediction error for the number of parameters shown on the horizontal axis.
- The prediction error is the sum of the squares of the differences between the validation data output and the model one-step-ahead predicted output.
- nk is the delay.

Three rectangles are highlighted on the plot in green, blue, and red. Each color indicates a type of best-fit criterion, as follows:

- Red — Best fit minimizes the sum of the squares of the difference between the validation data output and the model output. This rectangle indicates the overall best fit.
- Green — Best fit minimizes Rissanen MDL criterion.
- Blue — Best fit minimizes Akaike AIC criterion.

5 Results

5.1 Analyzed data

ECG plot

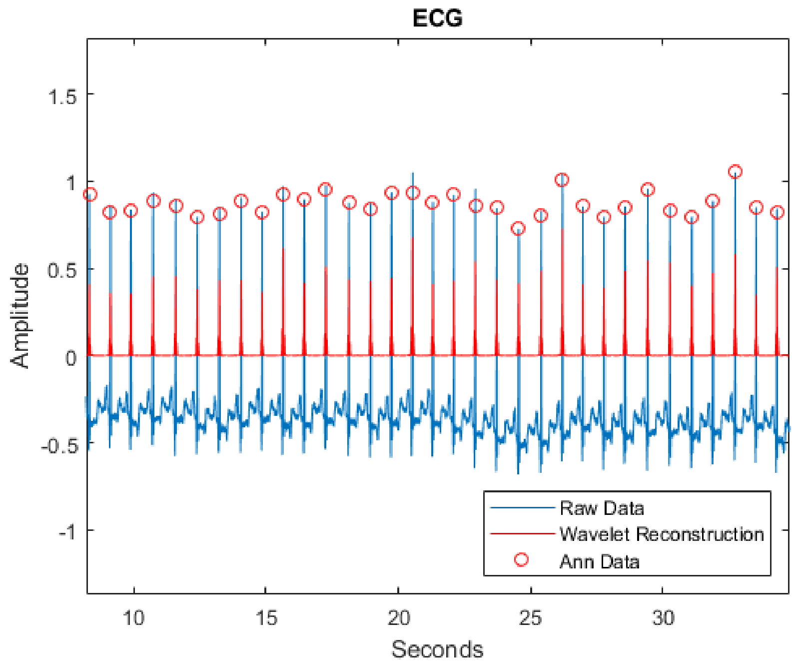

Figure 14: ECG plot patient No.100

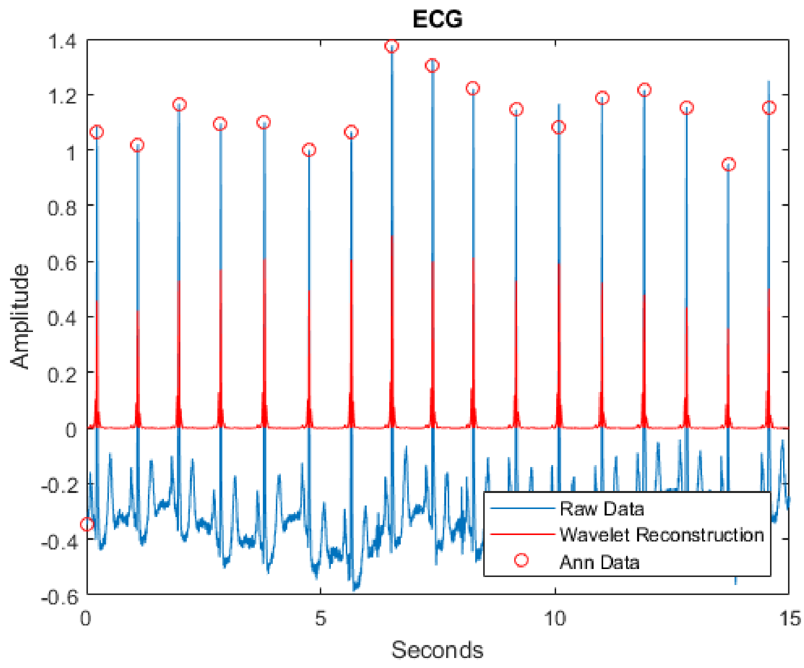

Figure 15: ECG plot patient No.101

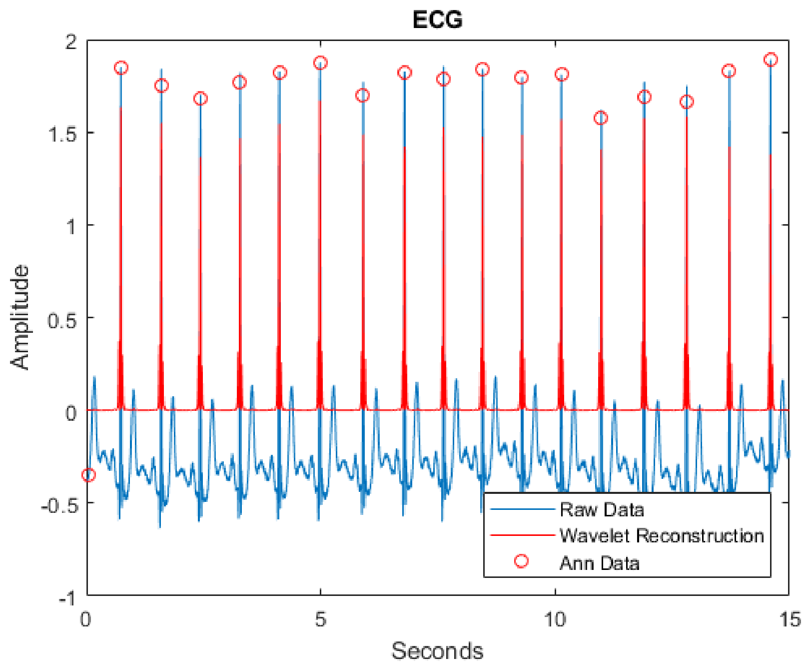

Figure 16: ECG plot patient No.103

Comparison of RRI of wavelet and ann

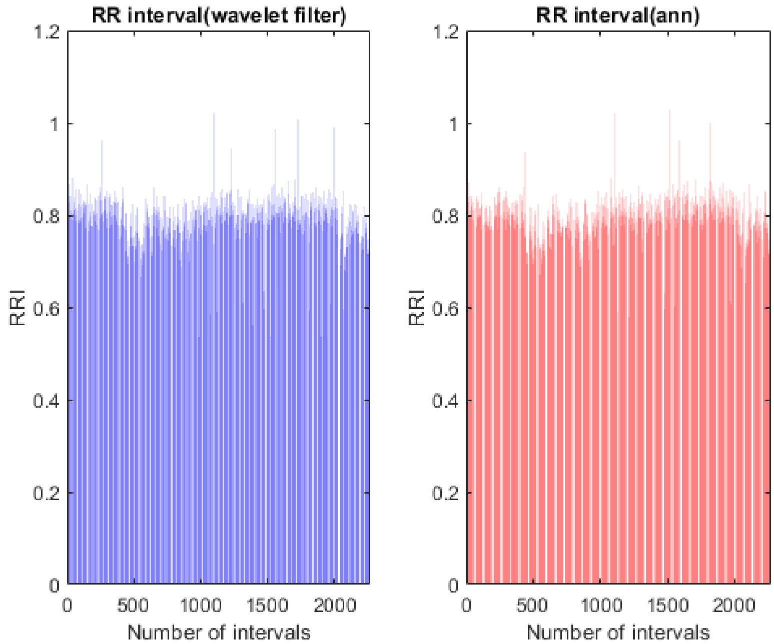

Figure 17: Comparison RRI of patient No.100

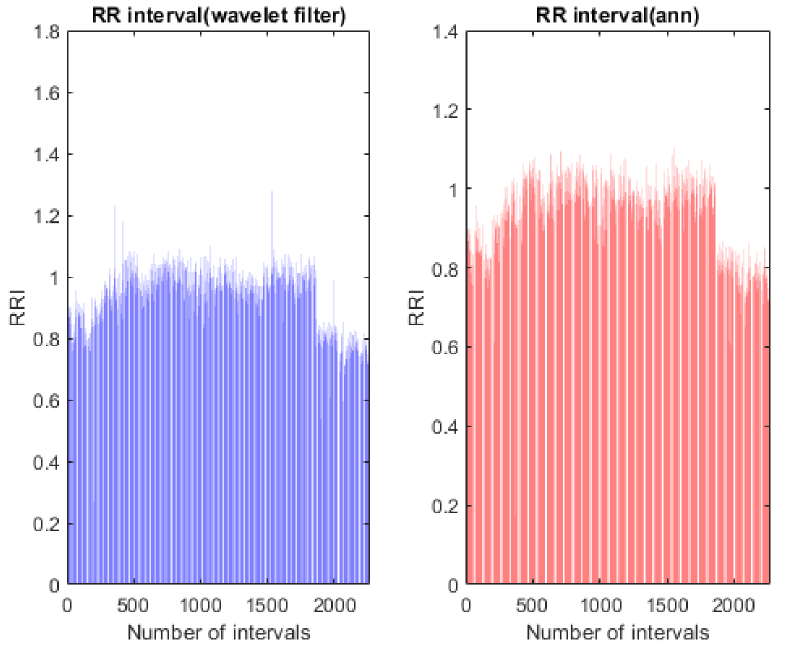

Figure 18: Comparison RRI of patient No.101

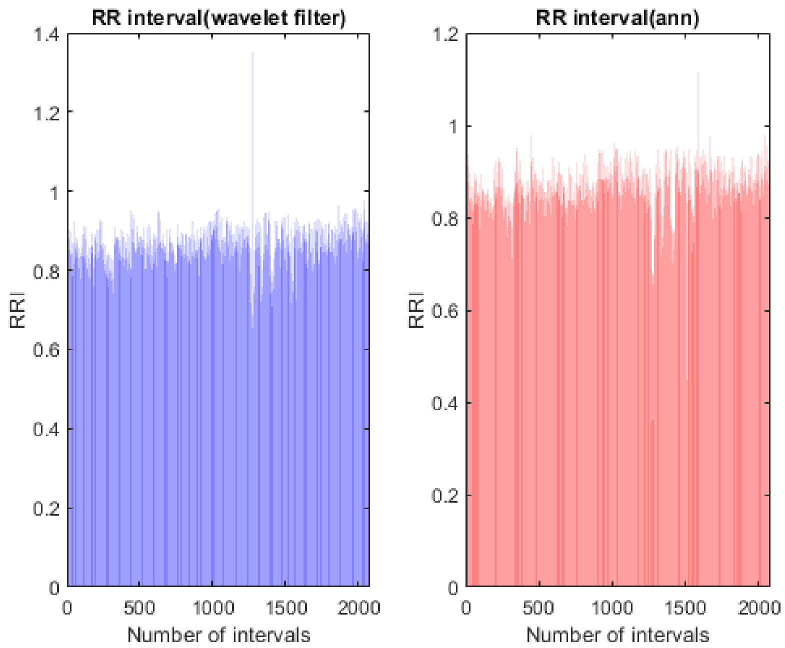

Figure 19: Comparison RRI of patient No.103

RRI(wavelet filter)

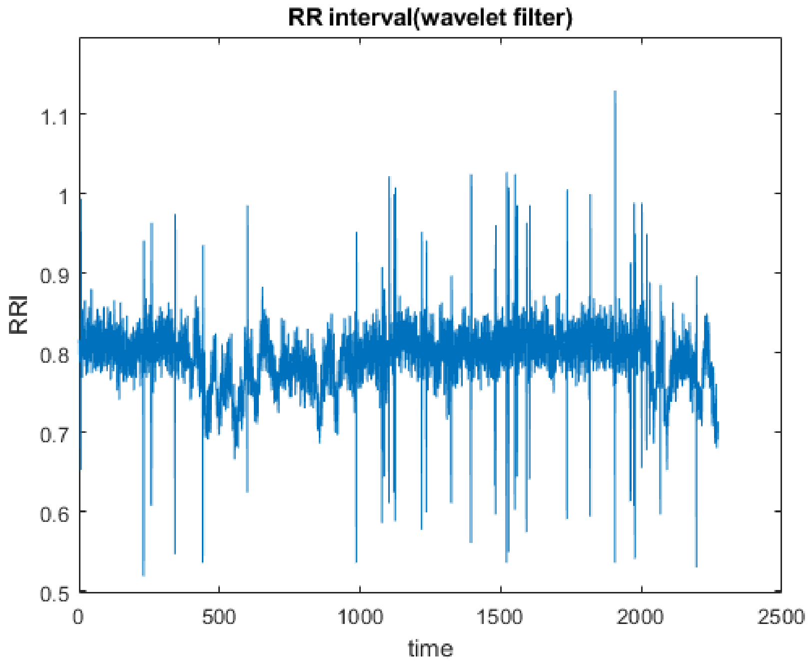

Figure 20: RRI of patient No.100

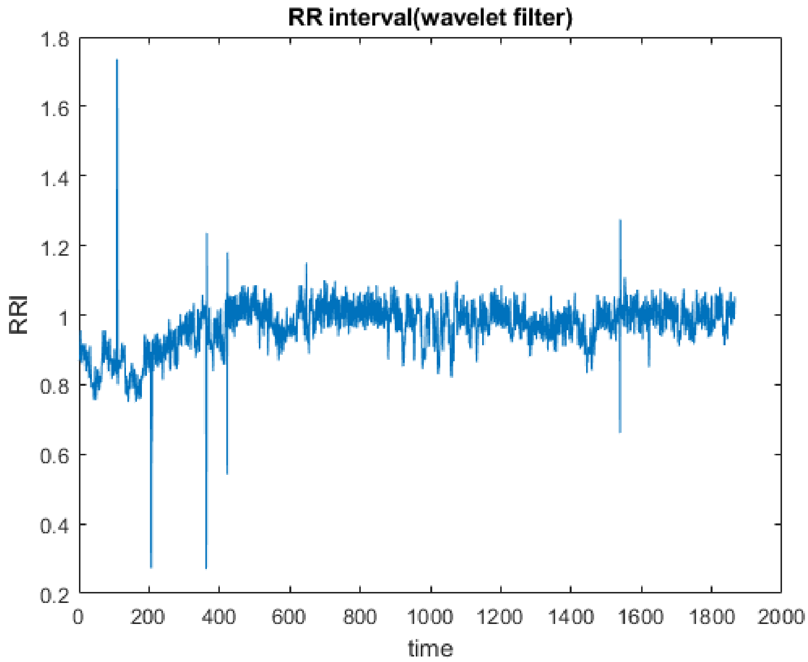

Figure 21: RRI of patient No.101

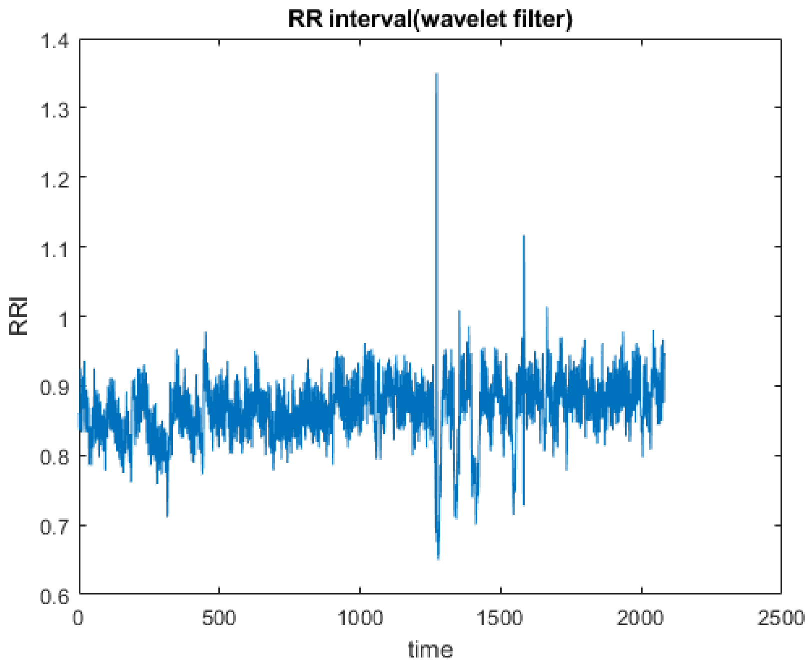

Figure 22: RRI of patient No.103

Random noise use for the input

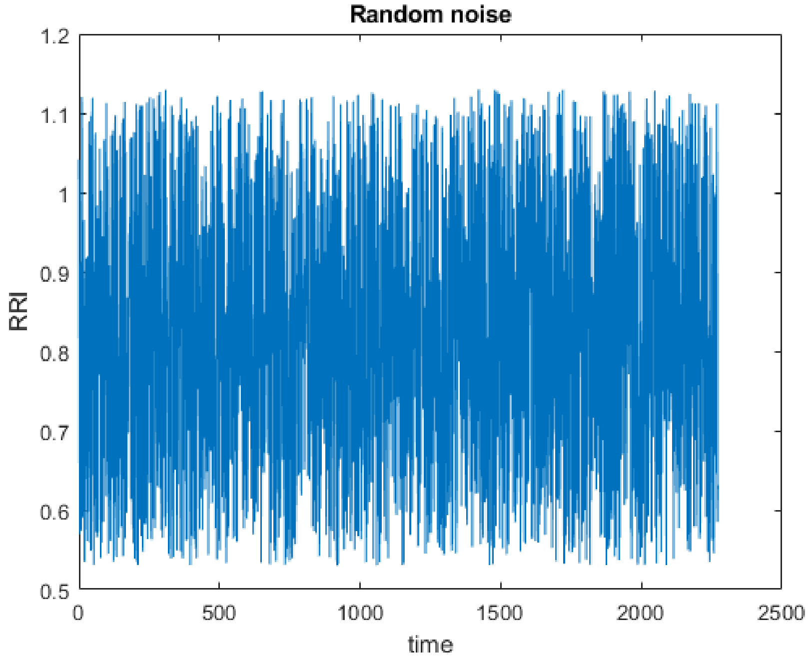

Figure 23: Random input for patient No.100

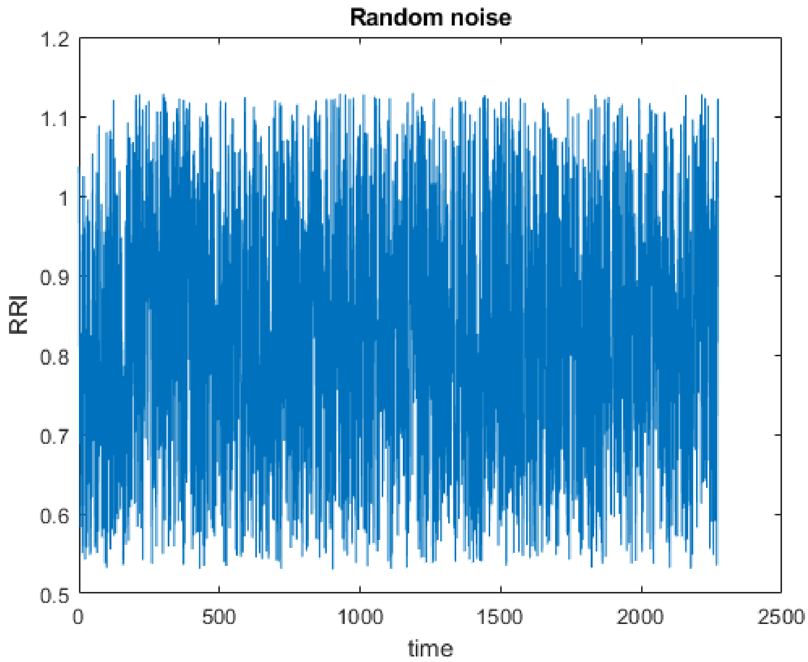

Figure 24: Random input for patient No.101

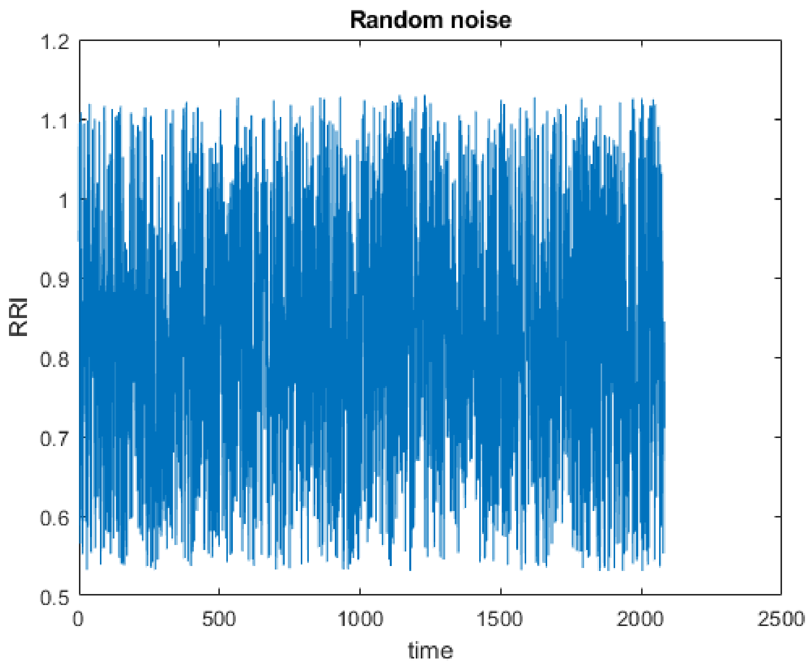

Figure 25: Random input for patient No.103

5.2 Models obtained

Patient No.100

ARX simulation

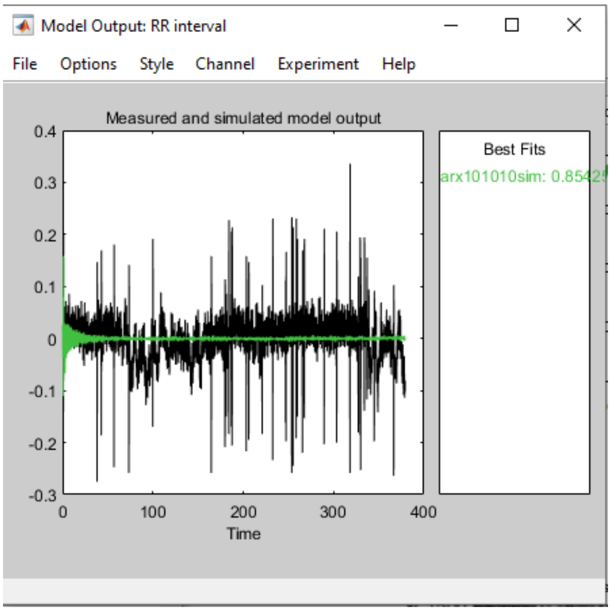

Figure 26: ARX simulation [10 10 10] patient No.100

```
arx101010sim =
Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)

  A(z) = 1 + 0.1704 (+/- 0.02105) z^-1 - 0.04468 (+/- 0.02063) z^-2
             + 0.02529 (+/- 0.01786) z^-3 + 0.04073 (+/- 0.016) z^-4
             - 0.01047 (+/- 0.01574) z^-5 - 0.1406 (+/- 0.01574) z^-6
             - 0.3779 (+/- 0.01601) z^-7 - 0.4907 (+/- 0.01787) z^-8
           - 0.2615 (+/- 0.02065) z^-9 + 0.09269 (+/- 0.02107) z^-10


  B(z) = -0.005468 (+/- 0.005245) z^-10
    -0.0008351 (+/- 0.005145) z^-11
    + 0.007651 (+/- 0.004888) z^-12 + 0.002122 (+/- 0.004645) z^-13
    - 0.0005287 (+/- 0.004614) z^-14 - 0.003943 (+/- 0.004615) z^-15
    + 0.00169 (+/- 0.004647) z^-16 + 0.003402 (+/- 0.004893) z^-17
    + 0.003545 (+/- 0.005153) z^-18 - 0.00517 (+/- 0.005258) z^-19

Name: arx101010sim
Sample time: 0.167 seconds

Parameterization:
   Polynomial orders:   na=10   nb=10   nk=10
   Number of free coefficients: 20
```

```
    Use "polydata", "getpvec", "getcov"for parameters and their
    uncertainties.

Status:
Estimated using ARX on time domain data "mydatad".
Fit to estimation data: 0.8543% (simulation focus)
FPE: 0.001775, MSE: 0.002344
More information in model's "Report" property.
```

ARX prediction

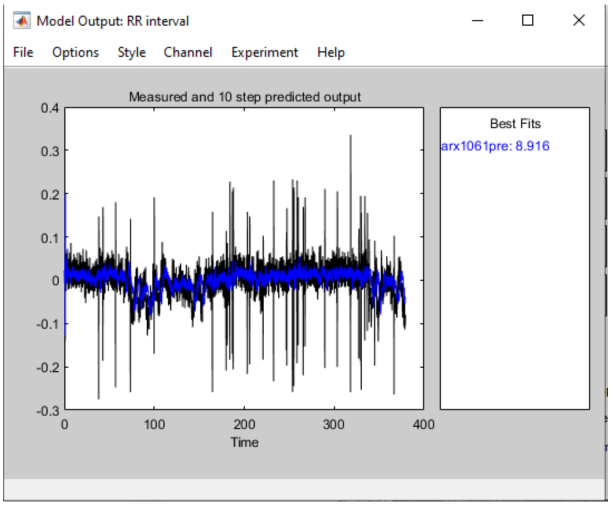

Figure 27: ARX prediction [10 6 1] patient No.100

```
arx1061pre =
Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)

  A(z) = 1 + 0.1341 (+/- 0.02111) z^-1 + 0.01188 (+/- 0.0208) z^-2
           + 0.02289 (+/- 0.01945) z^-3 - 0.001412 (+/- 0.01853) z^-4
           - 0.05048 (+/- 0.01832) z^-5 - 0.1407 (+/- 0.01833) z^-6
           - 0.2797 (+/- 0.01857) z^-7 - 0.3406 (+/- 0.01946) z^-8
           - 0.2127 (+/- 0.02075) z^-9 - 0.003087 (+/- 0.02101) z^-10


  B(z) = 0.005874 (+/- 0.004952) z^-1 + 0.001998 (+/- 0.004953) z^-2
    - 0.0001366 (+/- 0.004943) z^-3 + 0.0006895 (+/- 0.004943) z^-4
    + 0.006174 (+/- 0.004955) z^-5 + 0.004014 (+/- 0.004955) z^-6

Name: arx1061pre
Sample time: 0.167 seconds

Parameterization:
    Polynomial orders:   na=10   nb=6   nk=1
    Number of free coefficients: 16
```

```
    Use "polydata", "getpvec", "getcov" for parameters and their
    uncertainties.

Status:
Estimated using ARX on time domain data "mydatad".
Fit to estimation data: 17.55% (prediction focus)
FPE: 0.001659, MSE: 0.001621
More information in model's "Report" property.
```

ARMAX simulation

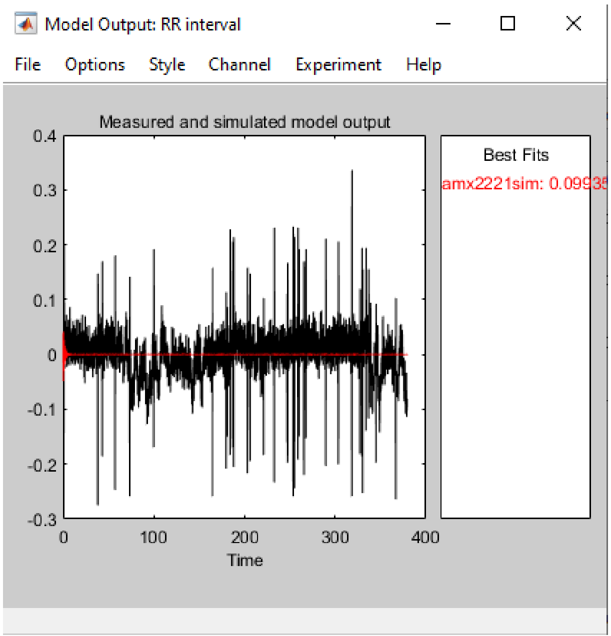

Figure 28: ARMAX simulation [2 2 2 1] patient No.100

```
amx2221sim =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 + 1.466 (+/- 0.4641) z^-1 + 0.8258 (+/- 0.4555) z^-2

  B(z) = 0.001892 (+/- 0.00366) z^-1 + 0.001775 (+/- 0.003763) z^-2

  C(z) = 1 + 1.477 (+/- 0.01093) z^-1 + 0.8531 (+/- 0.01093) z^-2

Name: amx2221sim
Sample time: 0.167 seconds

Parameterization:
    Polynomial orders:   na=2   nb=2   nc=2   nk=1
    Number of free coefficients: 6
    Use "polydata", "getpvec", "getcov" for parameters and their
    uncertainties.
```

```
Status:
Termination conditions for measured dynamics model
Near (local) minimum, (norm(g) < tol)..
Number of iterations: 5, Number of function evaluations: 14

Termination conditions for noise model
Near (local) minimum, (norm(g) < tol)..
Number of iterations: 4, Number of function evaluations: 10

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 0.008573% (simulation focus)
FPE: 0.00239, MSE: 0.002385
More information in model's "Report" property.
```

ARXMAX prediction

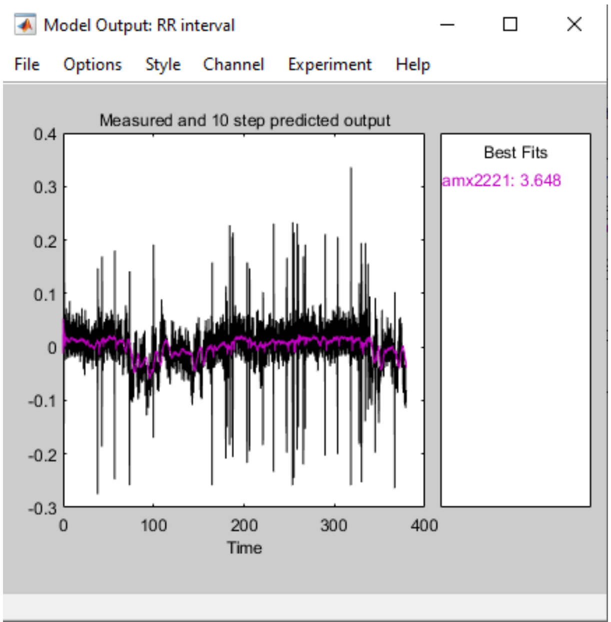

Figure 29: ARMAX prediction [2 2 2 1] patient No.100

```
amx2221pre =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 - 0.1308 (+/- 0.1302) z^-1 - 0.8464 (+/- 0.129) z^-2

  B(z) = 0.003205 (+/- 0.00381) z^-1 + 0.001042 (+/- 0.003856) z^-2

  C(z) = 1 - 0.08357 (+/- 0.139) z^-1 - 0.764 (+/- 0.1299) z^-2

Name: amx2221pre
Sample time: 0.167 seconds
```

```
Parameterization:
   Polynomial orders:   na=2   nb=2   nc=2   nk=1
   Number of free coefficients: 6
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.

Status:
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 8, Number of function evaluations: 19

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 7.466% (prediction focus)
FPE: 0.002053, MSE: 0.002042
More information in model's "Report" property.
```

BJ simulation

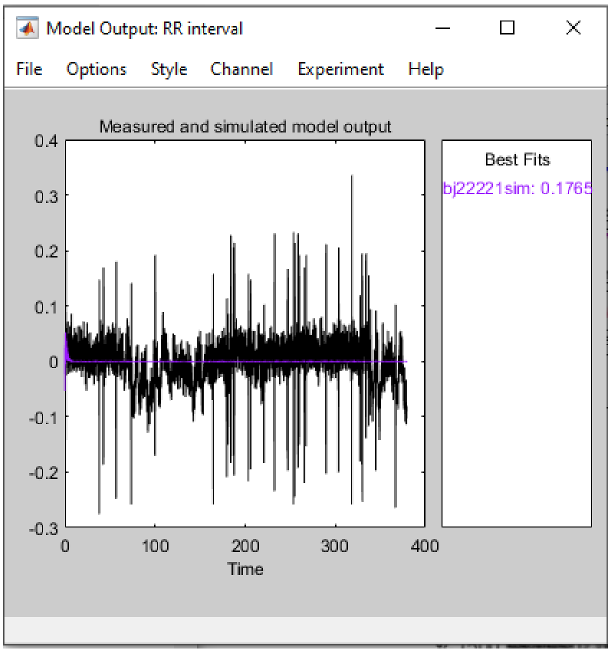

Figure 30: Box-Jenkins simulation [2 2 2 2 1] patient No.100

```
bj22221sim =
Discrete-time BJ model: y(t) = [B(z)/F(z)]u(t) + [C(z)/D(z)]e(t)
  B(z) = 0.001892 (+/- 0.00366) z^-1 + 0.001775 (+/- 0.003763) z^-2

  C(z) = 1 - 1.758 (+/- 0.02196) z^-1 + 0.8443 (+/- 0.02006) z^-2

  D(z) = 1 - 1.644 (+/- 0.03172) z^-1 + 0.6707 (+/- 0.03142) z^-2

  F(z) = 1 + 1.466 (+/- 0.4641) z^-1 + 0.8258 (+/- 0.4555) z^-2
```

```
Name: bj22221sim
Sample time: 0.167 seconds

Parameterization:
    Polynomial orders:   nb=2   nc=2   nd=2   nf=2   nk=1
    Number of free coefficients: 8
    Use "polydata", "getpvec", "getcov" for parameters and their
    uncertainties.

Status:
Termination conditions for measured dynamics model
Near (local) minimum, (norm(g) < tol)..
Number of iterations: 5, Number of function evaluations: 14

Termination conditions for noise model
Near (local) minimum, (norm(g) < tol)..
Number of iterations: 15, Number of function evaluations: 31

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 0.008573% (simulation focus)
FPE: 0.001948, MSE: 0.002385
More information in model's "Report" property.
```

BJ prediction

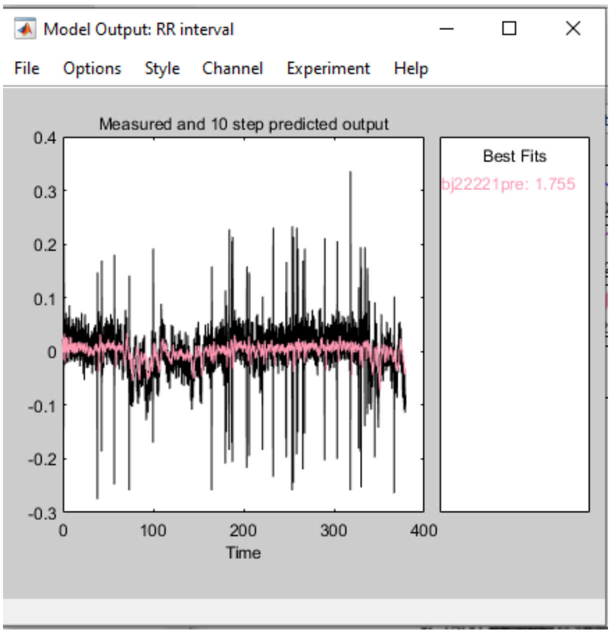

Figure 31: Box-Jenkins prediction [2 2 2 2 1] patient No.100

```
bj22221pre =
```

```
Discrete-time BJ model: y(t) = [B(z)/F(z)]u(t) + [C(z)/D(z)]e(t)

 B(z) = 0.0006135 (+/- 0.001507) z^-1 + 0.0004935 (+/- 0.001523) z^-2

  C(z) = 1 - 1.754 (+/- 0.02251) z^-1 + 0.8408 (+/- 0.02057) z^-2

  D(z) = 1 - 1.637 (+/- 0.03219) z^-1 + 0.6639 (+/- 0.03189) z^-2

  F(z) = 1 + 1.44 (+/- 0.08544) z^-1 + 0.9774 (+/- 0.08436) z^-2

Name: bj22221pre
Sample time: 0.167 seconds

Parameterization:
   Polynomial orders:   nb=2   nc=2   nd=2   nf=2   nk=1
   Number of free coefficients: 8
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.

Status:
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 9, Number of function evaluations: 29

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 9.931% (prediction focus)
FPE: 0.001948, MSE: 0.001935
More information in model's "Report" property.
```

Patient No.101

ARX simulation

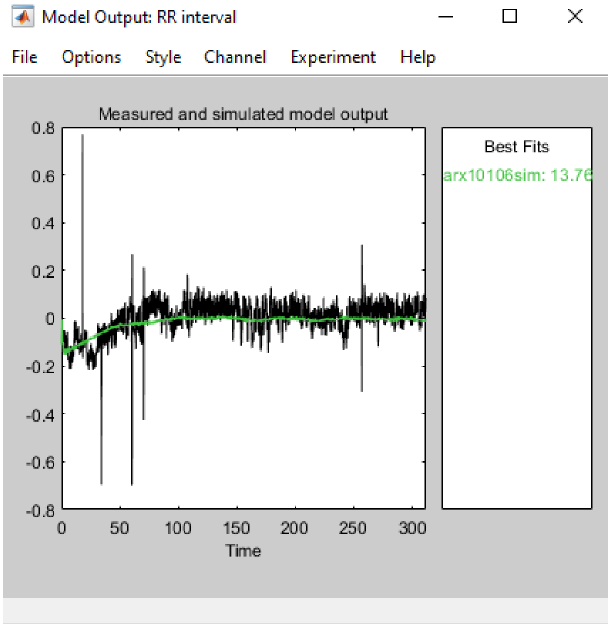

Figure 32: ARX simulation [10 10 6] patient No.101

```
arx10106sim =
Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)

  A(z) = 1 - 0.7594 (+/- 0.02314) z^-1 + 0.08561 (+/- 0.02823) z^-2
         - 0.06185 (+/- 0.02824) z^-3 - 0.08043 (+/- 0.02825) z^-4
         - 0.1844 (+/- 0.02794) z^-5 - 0.1966 (+/- 0.02794) z^-6
         - 0.06462 (+/- 0.02826) z^-7 + 0.08946 (+/- 0.02826) z^-8
         + 0.3094 (+/- 0.02823) z^-9 - 0.131 (+/- 0.02315) z^-10


  B(z) = -0.002288 (+/- 0.007063) z^-6 + 0.01986 (+/- 0.007443) z^-7
  - 0.01463 (+/- 0.00745) z^-8 + 0.0004233 (+/- 0.007415) z^-9
  + 0.005577 (+/- 0.007335) z^-10 + 0.002286 (+/- 0.007333) z^-11
  - 0.002399 (+/- 0.007414) z^-12 + 0.004433 (+/- 0.007449) z^-13
  - 0.009487 (+/- 0.007459) z^-14 + 0.001616 (+/- 0.007087) z^-15

Name: arx10106sim
Sample time: 0.167 seconds

Parameterization:
   Polynomial orders:   na=10   nb=10   nk=6
   Number of free coefficients: 20
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.
```

```
Status:
Estimated using ARX on time domain data "mydatad".
Fit to estimation data: 13.76% (simulation focus)
FPE: 0.00341, MSE: 0.004227
More information in model's "Report" property.
```

ARX prediction

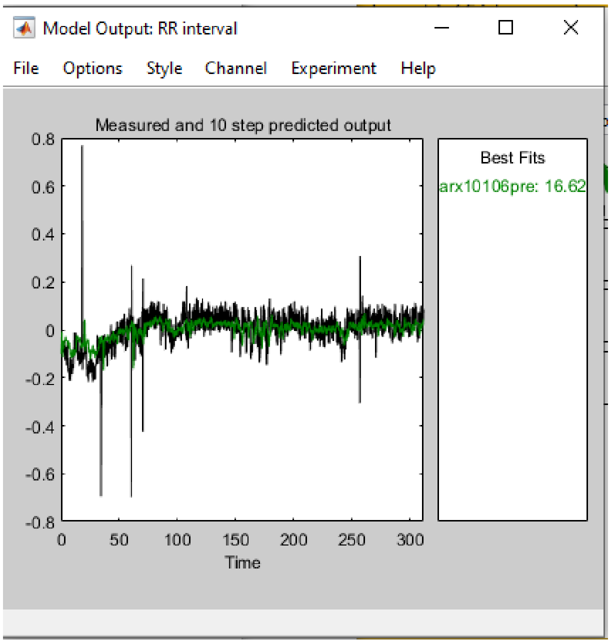

Figure 33: ARX prediction [10 10 6] patient No.101

```
arx10106pre =
Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)

  A(z) = 1 - 0.3989 (+/- 0.02331) z^-1 - 0.02681 (+/- 0.02487) z^-2
        - 0.04151 (+/- 0.02488) z^-3 - 0.06331 (+/- 0.02476) z^-4
        - 0.1362 (+/- 0.02448) z^-5 - 0.1718 (+/- 0.02447) z^-6
        - 0.1134 (+/- 0.02475) z^-7 - 0.006533 (+/- 0.02487) z^-8
        + 0.1428 (+/- 0.02484) z^-9 - 0.04775 (+/- 0.02327) z^-10


  B(z) = -0.001066 (+/- 0.007232) z^-6 + 0.02019 (+/- 0.007235) z^-7
    - 0.009534 (+/- 0.007247) z^-8 - 0.004817 (+/- 0.007253) z^-9
    + 0.003133 (+/- 0.007256) z^-10 + 0.005126 (+/- 0.007252) z^-11
    + 0.004312 (+/- 0.007253) z^-12 + 0.009354 (+/- 0.007257) z^-13
    - 0.004493 (+/- 0.007257) z^-14 - 0.0005182 (+/- 0.007259) z^-15

Name: arx10106pre
Sample time: 0.167 seconds
```

```
Parameterization:
   Polynomial orders:   na=10   nb=10   nk=6
   Number of free coefficients: 20
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.

Status:
Estimated using ARX on time domain data "mydatad".
Fit to estimation data: 30.25% (prediction focus)
FPE: 0.002855, MSE: 0.002765
More information in model's "Report" property.
```

ARMAX simulation

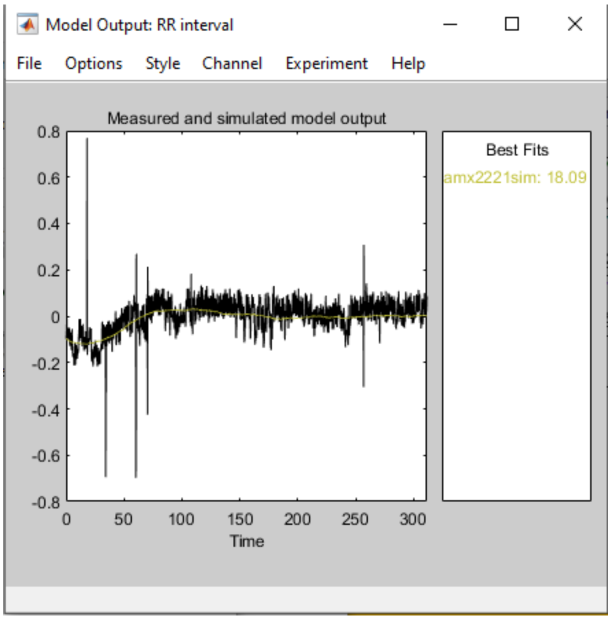

Figure 34: ARMAX simulation [2 2 2 1] patient No.101

```
amx2221sim =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 - 1.993 (+/- 0.00181) z^-1 + 0.9933 (+/- 0.001806) z^-2

  B(z) = -0.003414 (+/- 0.002798) z^-1 + 0.003431 (+/- 0.002788) z^-2

  C(z) = 1 - 1.237 (+/- 0.02361) z^-1 + 0.2371 (+/- 0.02365) z^-2

Name: amx2221sim
Sample time: 0.167 seconds

Parameterization:
   Polynomial orders:   na=2   nb=2   nc=2   nk=1
```

```
    Number of free coefficients: 6
    Use "polydata", "getpvec", "getcov" for parameters and their
    uncertainties.

Status:
Termination conditions for measured dynamics model
Near (local) minimum, (norm(g) < tol)..
Number of iterations: 13, Number of function evaluations: 35

Termination conditions for noise model
No improvement along the search direction with line search..
Number of iterations: 19, Number of function evaluations: 356

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 18.09% (simulation focus)
FPE: 0.003755, MSE: 0.003813
More information in model's "Report" property.
```

ARXMAX prediction

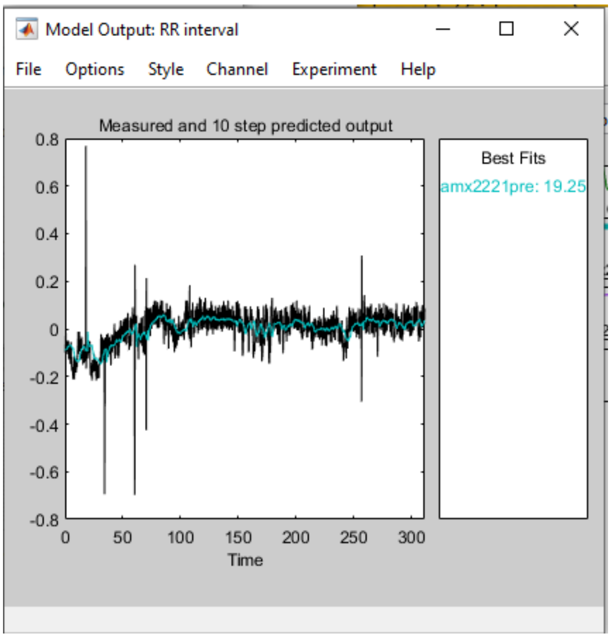

Figure 35: ARMAX prediction [2 2 2 1] patient No.101

```
amx2221pre =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 - 1.153 (+/- 0.07968) z^-1 + 0.1578 (+/- 0.07878) z^-2

  B(z) = 0.001955 (+/- 0.00725) z^-1 - 0.0001451 (+/- 0.007249) z^-2

  C(z) = 1 - 0.7472 (+/- 0.07938) z^-1 - 0.1625 (+/- 0.06808) z^-2
```

```
Name: amx2221pre
Sample time: 0.167 seconds

Parameterization:
   Polynomial orders:   na=2   nb=2   nc=2   nk=1
   Number of free coefficients: 6
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.

Status:
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 5, Number of function evaluations: 11

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 28.05% (prediction focus)
FPE: 0.002961, MSE: 0.002942
More information in model's "Report" property.
```

BJ simulation

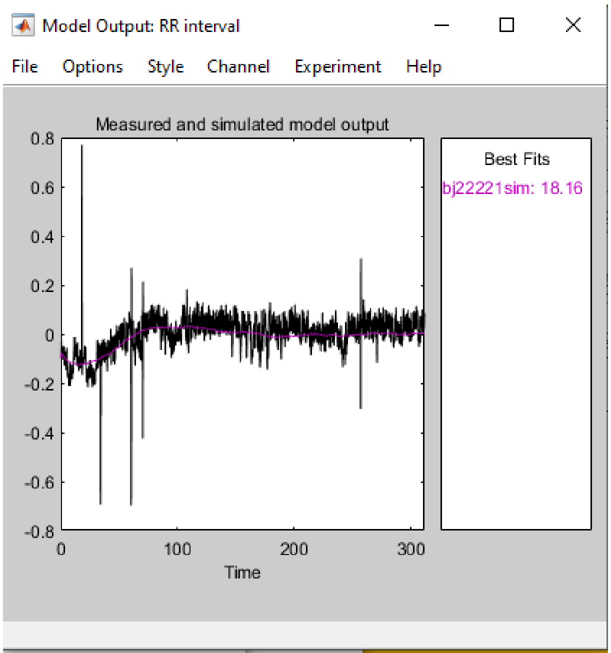

Figure 36: Box-Jenkins simulation [2 2 2 2 1] patient No.101

```
bj22221sim =
Discrete-time BJ model: y(t) = [B(z)/F(z)]u(t) + [C(z)/D(z)]e(t)
  B(z) = -0.003414 (+/- 0.002798) z^-1 + 0.003431 (+/- 0.002788) z^-2

  C(z) = 1 - 0.7315 (+/- 0.08497) z^-1 - 0.167 (+/- 0.06837) z^-2
```

```
  D(z) = 1 - 1.126 (+/- 0.08547) z^-1 + 0.1469 (+/- 0.08092) z^-2

  F(z) = 1 - 1.993 (+/- 0.00181) z^-1 + 0.9933 (+/- 0.001806) z^-2

Name: bj22221sim
Sample time: 0.167 seconds

Parameterization:
   Polynomial orders:   nb=2   nc=2   nd=2   nf=2   nk=1
   Number of free coefficients: 8
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.

Status:
Termination conditions for measured dynamics model
Near (local) minimum, (norm(g) < tol)..
Number of iterations: 13, Number of function evaluations: 35

Termination conditions for noise model
Near (local) minimum, (norm(g) < tol)..
Number of iterations: 6, Number of function evaluations: 20

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 18.16% (simulation focus)
FPE: 0.002941, MSE: 0.003806
More information in model's "Report" property.
```

BJ prediction

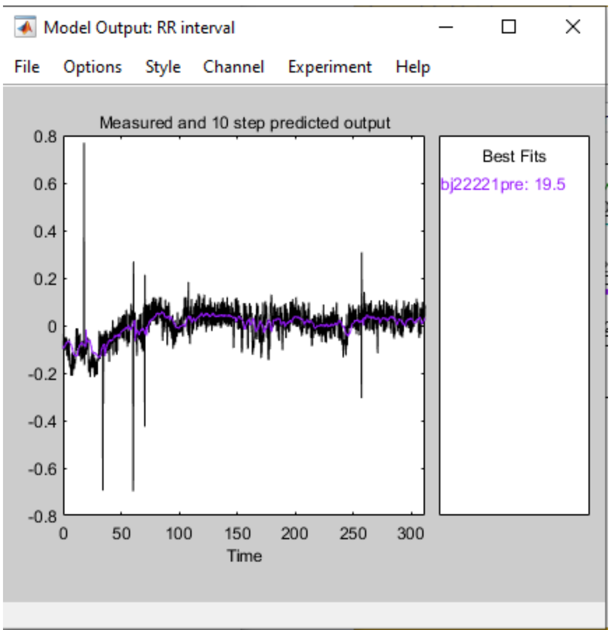

Figure 37: Box-Jenkins prediction [2 2 2 2 1] patient No.101

```
bj22221pre =
Discrete-time BJ model: y(t) = [B(z)/F(z)]u(t) + [C(z)/D(z)]e(t)
  B(z) = 0.0004762 (+/- 0.006354) z^-1 + 0.005194 (+/- 0.005379) z^-2

  C(z) = 1 - 0.7593 (+/- 0.07875) z^-1 - 0.1554 (+/- 0.06773) z^-2

  D(z) = 1 - 1.162 (+/- 0.0789) z^-1 + 0.1702 (+/- 0.07758) z^-2

  F(z) = 1 - 0.4853 (+/- 1.135) z^-1 - 0.5147 (+/- 1.135) z^-2

Name: bj22221pre
Sample time: 0.167 seconds

Parameterization:
   Polynomial orders:   nb=2   nc=2   nd=2   nf=2   nk=1
   Number of free coefficients: 8
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.

Status:
Termination condition: No improvement along the search direction with
line search..
Number of iterations: 15, Number of function evaluations: 256

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 28.06% (prediction focus)
FPE: 0.002967, MSE: 0.002941
More information in model's "Report" property.
```

Patient No.103

ARX simulation

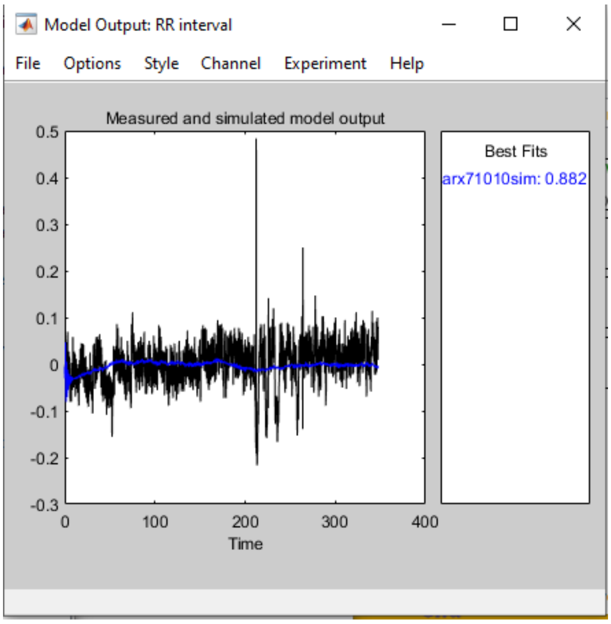

Figure 38: ARX simulation [7 10 10] patient No.103

```
arx71010sim =
Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)

  A(z) = 1 - 1.106 (+/- 0.02206) z^-1 + 0.5439 (+/- 0.03223) z^-2
           - 0.1376 (+/- 0.03435) z^-3 + 0.04813 (+/- 0.03448) z^-4
           - 0.05268 (+/- 0.03438) z^-5 - 0.2951 (+/- 0.03228) z^-6
                                          + 0.008487 (+/- 0.02215) z^-7


  B(z) = -0.0007369 (+/- 0.004733) z^-10
  + 0.005805 (+/- 0.005263) z^-11
   - 0.0008403 (+/- 0.005266) z^-12 - 0.002109 (+/- 0.00507) z^-13
   - 0.003192 (+/- 0.005037) z^-14 + 0.001238 (+/- 0.005039) z^-15
   - 0.004332 (+/- 0.00507) z^-16 + 0.00585 (+/- 0.005266) z^-17
   - 0.009423 (+/- 0.005267) z^-18 + 0.001748 (+/- 0.004737) z^-19

Name: arx71010sim
Sample time: 0.167 seconds

Parameterization:
    Polynomial orders:   na=7   nb=10   nk=10
    Number of free coefficients: 17
    Use "polydata", "getpvec", "getcov" for parameters and their
    uncertainties.
```

```
Status:
Estimated using ARX on time domain data "mydatad".
Fit to estimation data: 0.882% (simulation focus)
FPE: 0.001555, MSE: 0.002194
More information in model's "Report" property.
```

ARX prediction

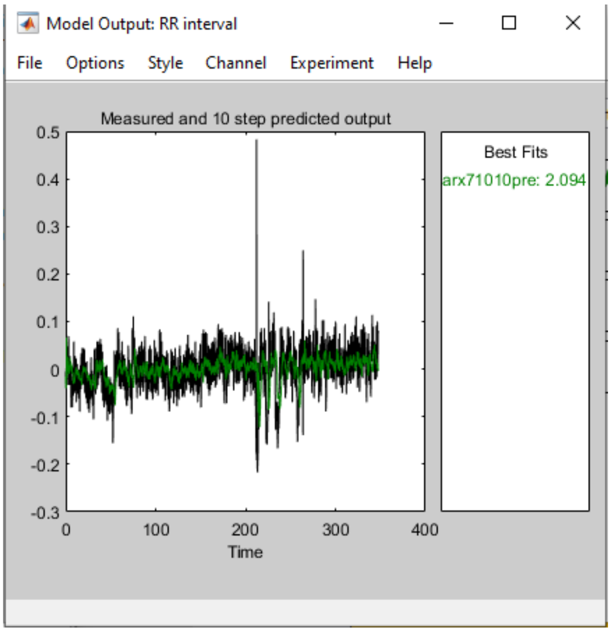

Figure 39: ARX prediction [7 10 10] patient No.103

```
arx71010pre =
Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)

  A(z) = 1 - 0.5203 (+/- 0.02112) z^-1 + 0.06802 (+/- 0.02398) z^-2
          - 0.02963 (+/- 0.02398) z^-3 + 0.0225 (+/- 0.024) z^-4
          - 0.05188 (+/- 0.02398) z^-5 - 0.0753 (+/- 0.02399) z^-6
                                            - 0.2874 (+/- 0.02114) z^-7


  B(z) = 0.002572 (+/- 0.004179) z^-10 + 0.004831 (+/- 0.004182) z^-11
   + 0.004568 (+/- 0.004186) z^-12 - 0.00119 (+/- 0.004185) z^-13
   - 0.0007825 (+/- 0.004182) z^-14 + 0.0007654 (+/- 0.004182) z^-15
   + 4.611e-05 (+/- 0.004186) z^-16 + 0.007764 (+/- 0.00419) z^-17
   - 0.002279 (+/- 0.004194) z^-18 + 0.004995 (+/- 0.004194) z^-19

Name: arx71010pre
Sample time: 0.167 seconds
```

```
Parameterization:
   Polynomial orders:    na=7    nb=10    nk=10
   Number of free coefficients: 17
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.

Status:
Estimated using ARX on time domain data "mydatad".
Fit to estimation data: 31.93% (prediction focus)
FPE: 0.001062, MSE: 0.001035
More information in model's "Report" property.
```

ARMAX simulation

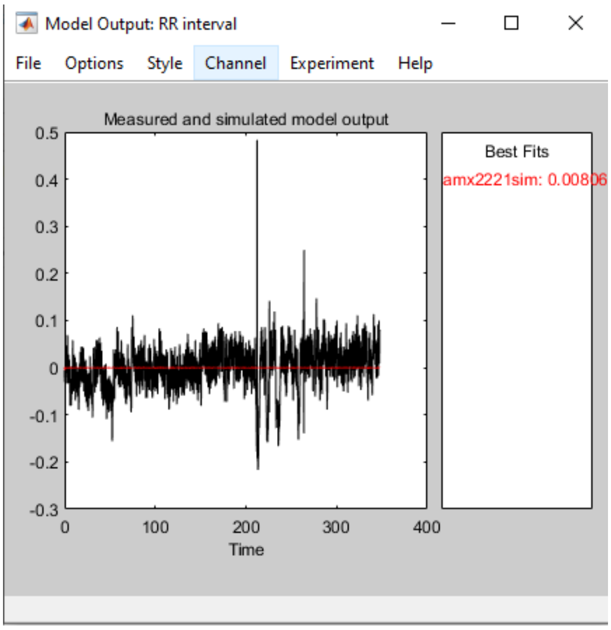

Figure 40: ARMAX simulation [2 2 2 1] patient No.103

```
amx2221sim =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 + 0.02451 (+/- 1.27) z^-1 + 0.4338 (+/- 1.093) z^-2

  B(z) = 0.0005909 (+/- 0.005426) z^-1 - 0.003145 (+/- 0.005979) z^-2

  C(z) = 1 + 0.6236 (+/- 0.01764) z^-1 + 0.5942 (+/- 0.01764) z^-2

Name: amx2221sim
Sample time: 0.167 seconds

Parameterization:
   Polynomial orders:    na=2    nb=2    nc=2    nk=1
```

```
   Number of free coefficients: 6
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.

Status:
Termination conditions for measured dynamics model
Near (local) minimum, (norm(g) < tol)..
Number of iterations: 6, Number of function evaluations: 15

Termination conditions for noise model
Near (local) minimum, (norm(g) < tol)..
Number of iterations: 4, Number of function evaluations: 9

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 0.007724% (simulation focus)
FPE: 0.001537, MSE: 0.002233
More information in model's "Report" property.
```

ARXMAX prediction

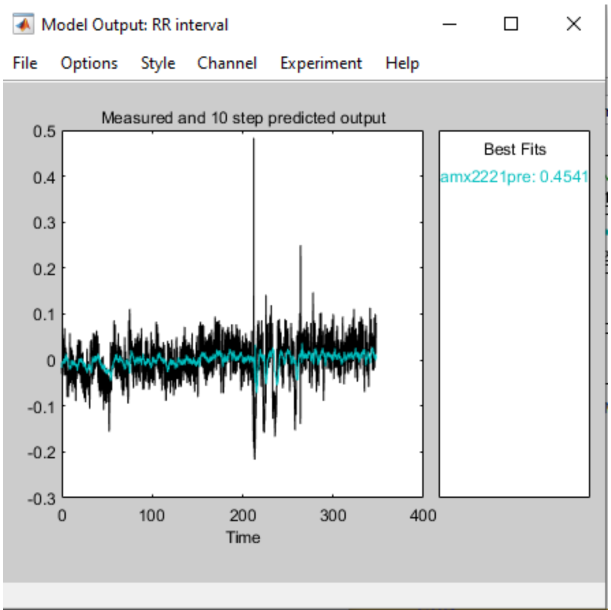

Figure 41: ARMAX prediction [2 2 2 1] patient No.103

```
amx2221pre =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 - 1.145 (+/- 0.07995) z^-1 + 0.1909 (+/- 0.06988) z^-2

  B(z) = 0.001689 (+/- 0.004347) z^-1 - 0.0008375 (+/- 0.004353) z^-2

  C(z) = 1 - 0.5303 (+/- 0.07885) z^-1 - 0.2193 (+/- 0.04045) z^-2
```

```
Name: amx2221pre
Sample time: 0.167 seconds

Parameterization:
   Polynomial orders:   na=2   nb=2   nc=2   nk=1
   Number of free coefficients: 6
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.

Status:
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 7, Number of function evaluations: 19

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 26.17% (prediction focus)
FPE: 0.001224, MSE: 0.001217
More information in model's "Report" property.
```

BJ simulation

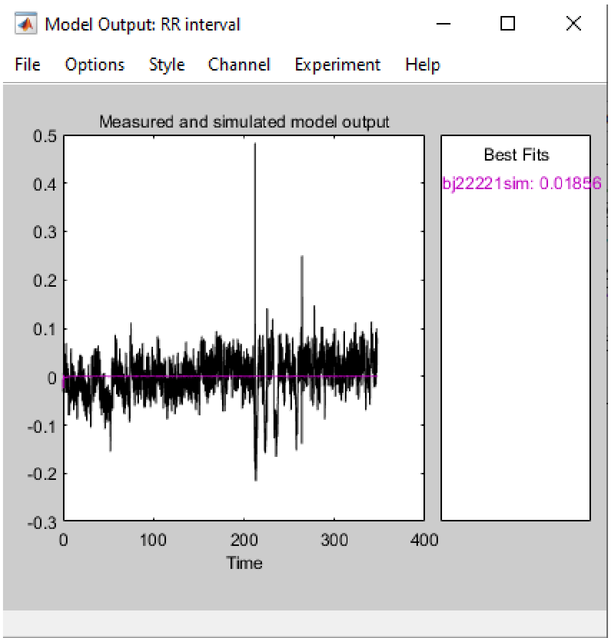

Figure 42: Box-Jenkins simulation [2 2 2 2 1] patient No.103

```
bj22221sim =
Discrete-time BJ model: y(t) = [B(z)/F(z)]u(t) + [C(z)/D(z)]e(t)
  B(z) = 0.0005909 (+/- 0.005426) z^-1 - 0.003145 (+/- 0.005979) z^-2

  C(z) = 1 - 0.5293 (+/- 0.0786) z^-1 - 0.22 (+/- 0.04043) z^-2

  D(z) = 1 - 1.144 (+/- 0.07974) z^-1 + 0.1896 (+/- 0.06979) z^-2
```

```
  F(z) = 1 + 0.02451 (+/- 1.27) z^-1 + 0.4338 (+/- 1.093) z^-2

Name: bj22221sim
Sample time: 0.167 seconds

Parameterization:
   Polynomial orders:   nb=2   nc=2   nd=2   nf=2   nk=1
   Number of free coefficients: 8
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.

Status:
Termination conditions for measured dynamics model
Near (local) minimum, (norm(g) < tol)..
Number of iterations: 6, Number of function evaluations: 15

Termination conditions for noise model
Near (local) minimum, (norm(g) < tol)..
Number of iterations: 14, Number of function evaluations: 54

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 0.007724% (simulation focus)
FPE: 0.001226, MSE: 0.002233
More information in model's "Report" property.
```

BJ prediction

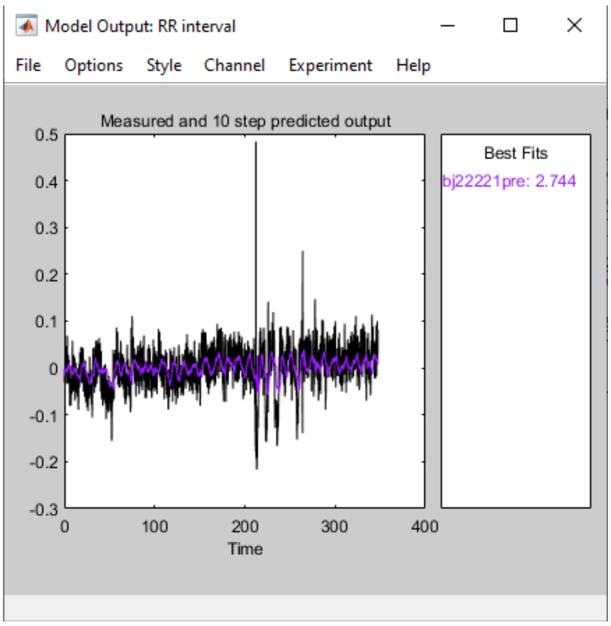

Figure 43: Box-Jenkins prediction [2 2 2 2 1] patient No.103

```
bj22221pre =
Discrete-time BJ model: y(t) = [B(z)/F(z)]u(t) + [C(z)/D(z)]e(t)
  B(z) = 0.001134 (+/- 0.001423) z^-1 - 0.000697 (+/- 0.001427) z^-2

  C(z) = 1 - 0.5438 (+/- 0.07736) z^-1 - 0.2216 (+/- 0.04066) z^-2

  D(z) = 1 - 1.15 (+/- 0.07842) z^-1 + 0.1958 (+/- 0.06816) z^-2

  F(z) = 1 - 1.991 (+/- 0.001354) z^-1 + 0.999 (+/- 0.001351) z^-2

Name: bj22221pre
Sample time: 0.167 seconds

Parameterization:
   Polynomial orders:   nb=2   nc=2   nd=2   nf=2   nk=1
   Number of free coefficients: 8
   Use "polydata", "getpvec", "getcov" for parameters and their
   uncertainties.

Status:
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 17, Number of function evaluations: 52

Estimated using PEM on time domain data "mydatad".
Fit to estimation data: 26.48% (prediction focus)
FPE: 0.001216, MSE: 0.001207
More information in model's "Report" property.
```

6 Conclusion

We analyzed the HRV data from PhysioNet using MATLAB.

At first,we obtained the RRI from these HRV data.

After obtained the RRI, we set as input random noise and set as output the RRI on the system identification app on MATLAB.

We tried three types of models,ARX,ARMAX and BJ.

As a result, prediction is better than simulation.However, models structures are not completely fit for these signal.

We have to increase the orders or look for the best orders(na,nb ... or nk) by using for-loop to reduce the errors.

Bibliografia

[1] PHYSIONET, https://physionet.org/physiotools/matlab/wfdb-app-matlab/

[2] WFDB Toolbox for MATLAB and Octave,https://archive.physionet.org/physiotools/matlab/wfdb-app-matlab/

[3] M. VALLVERDÚ, P. CAMINAL, *Dynamics Linear Systems Identification MATLAB Practices*, Signals and Biomedical System Division

[4] Data from this site, https://physionet.org/physiobank/database/mitdb/

[5] present(MATLAB syntax), https://jp.mathworks.com/help/ident/ref/present.html

[6] Identify Linear Models Using System Identification App, https://jp.mathworks.com/help/ident/gs/identify-linear-models-using-the-gui.html